Computational Mathematics and Mathematical Physics, Vol 42, No 9, 2002, pp 1372–1382
Translated from Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, Vol 42, No 9, 2002, pp 1425–1435

Recognition Algorithms with Representative Sets (Logic Algorithms)

Yu. I. Zhuravlev

Computing Center, Russian Academy of Sciences, ul. Vavilova 40, Moscow, GSP-1, 119991 Russia

e-mail: zhur@ccas.ru

Received January 10, 2002

Abstract—Algorithms based on the decomposition of classes by disjunctive normal forms and the choice of conjunction and pattern weights are investigated. An algorithm that is optimal under the sliding check is constructed. Techniques for reducing the dimensionality of the optimization problem when the control and training samples do not overlap are described.

1. BASIC DEFINITIONS, NOTATION, AND STATEMENT OF THE PROBLEM

We consider the problem of recognition or classification with two nonoverlapping classes K_1 and K_2. Objects are described by the sets of n binary features taking the values 0 or 1; $S = (\alpha_1, \ldots, \alpha_n)$, $\alpha_i \in \{0, 1\}$, $K_1 \cap K_2 = \varnothing$. Below, we assume that the reader is familiar with the terminology used in the mathematical theory of the representation and minimization of Boolean functions by disjunctive normal forms (DNF) [1–3]. The terminology of the algebraic theory of algorithms is also used [3–5].

The training (input) information $I_0(K_1, K_2)$ is given by the sets of objects

$$\tilde{K}_1 = \{S_1, \ldots, S_k\} \in K_1, \quad \tilde{K}_2 = \{S_{k+1}, \ldots, S_{k+l}\} \in K_2,$$

$$S_i = \{\alpha_{i1}, \ldots, \alpha_{in}\}, \quad \alpha_{ij} \in \{0, 1\}, \quad i = 1, 2, \ldots, k+l, \quad j = 1, 2, \ldots, n.$$

For every S, the recognizing algorithm $\mathcal{A}$ places S into one of the classes K_1 or K_2 using its description and $I_0(K_1, K_2)$ or rejects it. The algorithm $\mathcal{A}$ is the composition of an operator $\mathcal{B}$ and a decision rule $\mathcal{C}$. The operator $\mathcal{B}$ transforms $(I_0(K_1, K_2), S)$ to a pair of estimates Γ_1 and Γ_2 voting for the classes K_1 and K_2, respectively. The simplest decision rule $C(\Gamma_1(S), \Gamma_2(S))$ is used:

$$\Gamma_1(S) > \Gamma_2(S), \quad S \in K_1; \;\; \Gamma_1(S) < \Gamma_2(S), \quad S \in K_2;$$

$$\Gamma_1(S) = \Gamma_2(S), \text{ rejection.}$$

We also use the test information that includes the sets of objects

$$\tilde{K}'_1 = \{S'_1, \ldots, S'_m\} \in K_1, \quad \tilde{K}'_2 = \{S'_{m+1}, \ldots, S'_{m+q}\} \in K_2,$$

$$S'_i = \{\beta_{i1}, \ldots, \beta_{in}\}, \quad \beta_{ij} \in \{0.1\}, \quad i = 1, 2, \ldots m+q, \quad j = 1, 2, \ldots, n.$$

The training and test information is often specified in the form of training and test tables; therefore, this information can be designated by

$$[TO]_{n,k,l}, \quad [TK]_{n,m,q},$$

respectively. If the descriptions of the objects S_{i_1}, S_{i_2}, and S_{i_t} (or S'_{u_1}, S'_{u_2}, and S'_{u_v}, respectively) are withdrawn from the corresponding set, then we use the notation

$$[TO]_{n,k,l} \backslash \{S_{i_1}, S_{i_2}, S_{i_t}\} \quad \text{and} \quad [TK]_{n,m,q} \backslash \{S'_{u_1}, S'_{u_2}, S'_{u_v}\}.$$

If only S_i is withdrawn, we have $[TO]_{n,k,l} \backslash S_i = [TO]_{n,k-1,l}$ for $S_i \in K_1$ and $[TO]_{n,k,l-1}$ for $S_i \in K_2$. A similar notation is used for the test information.

The incompletely defined Boolean functions $F_1(x_1, \ldots, x_n)$ and $F_2(x_1, \ldots, x_n)$ are associated with the recognition problem: $F_1(x_1, \ldots, x_n)$ is equal to 1 on $\tilde{K}_1$, is equal to 0 on $\tilde{K}_2$, and is undefined for the other arguments. $F_2 = \bar{F}_1$ on $\tilde{K}_1 \cup \tilde{K}_2$ and is undefined for the other arguments.

Below, we consider families of algorithms $\{\mathcal{A}\}$ and, using $[TO]_{n,k,l}$ or the pair $[TO]_{n,k,l}$, $[TK]_{n,m,q}$, seek an algorithm maximizing one of the following objective functionals.

1. The sliding check functional $\varphi_{sl}(\mathcal{A})$. The algorithm

$$\mathcal{A}([TO]_{n,k,l} \backslash S_i), \quad i = 1, 2, \ldots, k+l$$

is used. If r is the number of recognized S_i $(i = 1, 2, \ldots, k + l)$, then $\varphi_{sl}(\mathcal{A}) = r/(k + l)$.

2. The independent check functional $\varphi_{in}(\mathcal{A})$. The algorithm

$$\mathcal{A}([TO]_{n,k,l} \backslash S'_j), \quad j = 1, 2, \ldots, m, m+l, \ldots, m+q$$

is used. If t is the number of recognized S'_j $(j = 1, 2, \ldots, m, m + 1, \ldots, m + q)$, then

$$\varphi_{in}(\mathcal{A}) = t/(m+q).$$

In what follows, we deal with the DNFs $\mathcal{D}(F)$ that realize incompletely defined Boolean functions F. Let $N_1(F)$ and $N_0(F)$ be the sets of units (respectively, zeros) of F and $\mathcal{D}(F) = Q_1 \vee \ldots \vee Q_p$, where Q_i are elementary conjunctions $(i = 1, 2, \ldots, p)$. Then

$$N_{Q_i} \cap N_1(F) \neq \varnothing, \quad N_{Q_i} \cap N_1(F) = \varnothing, \quad i = 1, 2, \ldots, p;$$

i.e., it is assumed that all elementary conjunctions in $\mathcal{D}(F)$ are admissible.

2. THE SET OF RECOGNIZING ALGORITHMS

The family of algorithms $\mathcal{A}$ defined below is a variant of the set of algorithms with representative sets [6]. It is determined by the DNFs

$$\mathcal{D}_1(F_1) = \bigvee_{i=1}^{t} Q_i, \quad \mathcal{D}_2(F_2) = \bigvee_{j=1}^{v} Q'_j, \tag{1}$$

parameters $W(Q_i)$ $(i = 1, 2, \ldots, t)$, $W(Q'_j)$ $(j = 1, 2, \ldots, v)$, and parameters $W(S_i) = W_i$ $(i = 1, 2, \ldots, k, k + 1, \ldots, q + l)$. In some cases, the notation

$$W(Q_i) = W^i, \quad i = 1, 2, \ldots, t, \quad W(Q'_j) = W^j, \quad j = 1, 2, \ldots, v$$

is used. The algorithm $\mathcal{A}$ is represented as a composition $\mathcal{B}\,\mathcal{C}$ of the recognizing operator $\mathcal{B}$ and the decision rule $\mathcal{C}$ described above. The operator $\mathcal{B}$ is determined, first, by specifying $\mathcal{D}(F_1)$ and $\mathcal{D}(F_2)$ in (1). Here, the elementary conjunctions Q_i and Q'_j $(i = 1, 2, \ldots, t, j = 1, 2, \ldots, v)$ are not necessarily simple implicants, i.e., the intervals N_{Q_i} and $N_{Q'_j}$ are not necessarily maximal. Second, by specifying a method for obtaining the estimates $\Gamma_1(Q, S)$, $\Gamma_2(Q, S)$, $\Gamma_1(S)$, and $\Gamma_2(S)$:

$$\begin{aligned} \Gamma_1(Q, S) &= \Bigg(\sum_{S_i \in \{N_1(F_1) \cap N_Q\}} Q(S_i)W(S_i)\Bigg)W(Q)Q(S), \quad \{N_1(F_1) \cap N_Q\} = \varnothing, \quad \Gamma_1(Q, S) = 0, \\ \Gamma_2(Q', S) &= \Bigg(\sum_{S_i \in \{N_1(F_2) \cap N_{Q'}\}} Q'(S_i)W(S_i)\Bigg)W(Q')Q(S), \quad \{N_1(F_2) \cap N_{Q'}\} = \varnothing, \quad \Gamma_2(Q, S) = 0; \end{aligned} \tag{2}$$

$$\Gamma_1(S) = \sum_{t=1}^{t}\Gamma_1(Q_t, S) = \left(\sum_{t=1}^{t} Q_t(S)W(Q_t) \sum_{S_u \in \{N_1(F_1)\cap N_Q\}} W(S_u)\right)Q(S),$$

$$\Gamma_2(S) = \sum_{i=1}^{v}\Gamma_1(Q'_j, S) = \left(\sum_{j=1}^{v} Q'_j(S)W(Q'_j) \sum_{S_u \in \left\{N_1(F_1)\cap N_{Q'_j}\right\}} W(S_u)\right)Q(S). \quad (3)$$

Evidently, $\Gamma_1(S)$ and $\Gamma_2(S)$ are bilinear forms

$$\sum d_{tt}W(Q_t)W(S_t), \quad \sum c_{ju}W(Q'_j)W(S_u). \quad (4)$$

When performing the sliding check, the objects S_t are extracted from $[TO]_{n,k,l}$. Therefore, (2) and (3) are replaced by (5) and (6), respectively:

$$\Gamma_1(Q, S_t) = \left(\sum_{S_t \in \{(N_1(F_1)\backslash S_t)\cap N_Q\}} Q(S_t)W(S_t)\right)W(Q)Q(S_t), \{(N_1(F_1)\backslash S_t)\cap N_Q\} = \varnothing, \Gamma_1(Q, S_t) = 0. \quad (5)$$

Similarly, $\Gamma_2(Q', S_t)$ is changed. At the same time,

$$\Gamma_2(S_t) = \sum_{v=1}^{t}\Gamma_1(Q_v, S_t) = \left(\sum_{v=1}^{t} Q_v(S_t)W(Q_v) \sum_{S_u \in \{(N_1(F_1)\backslash S_t)\cap N_Q\}}^{t} W(S_u)\right)Q_v(S_t). \quad (6)$$

Similarly, $\Gamma_2(S_t)$ is changed.

3. OPTIMIZATION OF ALGORITHMS WITH RESPECT TO THE SLIDING CHECK FUNCTIONAL $\varphi_{SL}(\mathcal{A})$

Let $\tilde{\gamma}^1 = (\gamma_{11}, \ldots, \gamma_{1j}, \ldots \gamma_{1n}), \ldots, \tilde{\gamma}^p = (\gamma_{p1}, \ldots, \gamma_{pj}, \ldots, \gamma_{pn})$ be the vertices of the n-dimensional unit cube E^n. Let N_Q be the minimal interval (if it exists) containing the vertices $\tilde{\gamma}^1, \ldots, \tilde{\gamma}^p$ and Q is the corresponding conjunction.

Obviously, such an interval exists if and only if the matrix $\|\gamma_{ij}\|_{n\times p}$ contains at least one zero or unit column.

In what follows, the conjunction Q is denoted by $Q(\tilde{\gamma}^1, \ldots, \tilde{\gamma}^p)$.

Let $j_1, \ldots, j_v$ be the indices of all identical columns in the matrix $\|\gamma_{ij}\|_{n\times p}$, where $v < p$.

Proposition 1. *The following relation holds*:

$$Q(\tilde{\gamma}^1, \ldots, \tilde{\gamma}^p) = x_{j_1}^{\gamma_{ij_1}} \ldots x_{j_v}^{\gamma_{ij_v}}, \quad i = 1, 2, \ldots, p.$$

Proof. It is clear that $Q(\tilde{\gamma}^i) = 1$ for $i = 1, 2, \ldots, p$. If an arbitrary factor is added to Q, the transformed conjunction in at least one $\tilde{\gamma}^u$ $(1 \le u \le p)$ will turn to zero. A deletion of any factor yields an elementary conjunction corresponding to a nonminimal interval containing the vertices $\tilde{\gamma}^1, \ldots, \tilde{\gamma}^p$. The proposition is proved.

Let $\{Q(S_u, S_v)\}$ $(u = 1, 2, \ldots, k, v = 1, 2, \ldots, u-1, u+1, \ldots, k)$ be a set of conjunctions with minimal corresponding intervals containing all possible pairs of different points from $\tilde{K}_1$. Similarly, $\{Q(S_p, S_d)\}$ $(S_p \neq S_d, S_p, S_d \in \tilde{K}_2)$ is defined.

Definition 1. An object S_t is called isolated in K_1 (respectively, in K_2) if every nonempty $N_{Q(S_u, S_v)}$, $Q(S_u, S_v) \in \{Q(S_u, S_v)\}$ (respectively, every $N_{Q(S_p, S_d)}$, $Q(S_p, S_d) \in \{Q(S_p, S_d)\}$) has a nonempty intersection with $N_0(F_1)$ (respectively, with $N_0(F_2)$).

Let us single out all isolated objects $S_{r_1}, \ldots, S_{r_p}$ ($S_{r_{p+1}}, \ldots, S_{r_{p+d}}$, respectively) in $\tilde{K}_1$ (in $\tilde{K}_2$, respectively). Consider the sets

$$M(\tilde{K}_1) = \tilde{K}_1 \backslash \{S_{r_1}, \ldots, S_{r_p}\}, \quad 1 \le r_1, \ldots, r_p \le k,$$

$$M(\tilde{K}_2) = \tilde{K}_2 \backslash \{S_{r_{p+1}}, \ldots, S_{r_{p+d}}\}, \quad k+1 \le r_{p+1}, \ldots, r_{p+d} \le k+l,$$

$$\{Q(S_i, S_j)\}, \quad S_i \ne S_j, \quad S_i, S_j \in M(\tilde{K}_1), \quad N_{Q(S_i, S_j)} \cap N_0(F_1) = \varnothing,$$

$$\{Q(S_u, S_v)\}, \quad S_u \ne S_v, \quad S_u, S_v \in M(\tilde{K}_2), \quad N_{Q(S_u, S_v)} \cap N_0(F_2) = \varnothing.$$

Consider an arbitrary algorithm $\mathcal{A}$ whose operator is determined by arbitrary DNFs

$$\mathfrak{D}_1(F_1) = \bigvee_{i=1}^{t} Q_i, \quad \mathfrak{D}_2(F_2) = \bigvee_{j=1}^{g} Q'_j,$$

possessing the following properties:

(a) For every $Q(S_i, S_j) \in \{Q(S_i, S_j)\}$, there exists a $Q_r \in \mathfrak{D}_1(F_1)$ ($1 \le r \le t$, $r = r(i, j)$) such that

$$N_{Q(S_i, S_j)} \subseteq N_{Q_r};$$

(b) For every $Q(S_u, S_v) \in \{Q(S_u, S_v)\}$, there exists a $Q_m \in \mathfrak{D}_2(F_2)$ ($1 \le m \le g$, $m = m(u, v)$) such that

$$N_{Q(S_u, S_v)} \subseteq N'_{Q_m}.$$

Below, such pairs of DNFs $(\mathfrak{D}_1(F_1), \mathfrak{D}_2(F_2))$ are called sufficient.

Assume, in addition, that all parameters $W(Q)$, $Q \in \mathfrak{D}_1(F_1)$, $W(Q')$, $Q' \in \mathfrak{D}_2(F_2)$, $W^1, \ldots, W^{k+l}$ are positive.

Theorem 1. *Any algorithm with a sufficient pair of* DNFs $(\mathfrak{D}_1(F_1), \mathfrak{D}_2(F_2))$ *and positive parameters correctly recognizes all nonisolated objects of the training sample in the sliding check mode.*

Proof. Let $S_i \in \tilde{K}_1$ and S_i be a nonisolated object. Then, there exists an object S_j in $K_1 \backslash S_i$ and there exists a conjunction Q_r ($1 \le i \le k$, $1 \le j \le k$, $i \ne j$) in $\mathfrak{D}_1(F_1)$ such that $N_{Q(S_i, S_j)}, \subseteq N_{Q_r}$.

We have $S_i, S_j \in N_{Q_r}$, $Q_r(S_i) = Q_r(S_j) = 1$, and $S_j \in \{N_1(F_1) \cap N_Q\} \backslash S_i$. Then, by virtue of (5) and (6), we have

$$\Gamma_1(Q_r, S_i) \ge W(Q_r) W(S_i) > 0, \quad \Gamma_1(S_i) > 0.$$

The decision rule places S_i into the class K_1, since none of the conjunctions Q'_j from $\mathfrak{D}_2(F_2)$ is equal to unity on S_i (here, Q'_j are the admissible elementary conjunctions for F_2). Similarly, one can prove that every nonisolated object S_u in $\tilde{K}_2$ will be assigned to the class $\tilde{K}_2$ by the algorithm $\mathcal{A}$. The theorem is proved.

Theorem 2. *Any algorithm with arbitrary* $\mathfrak{D}_1(F_1)$, $\mathfrak{D}_2(F_2)$, *and arbitrary parameters* $W(Q)$, $W(S_i)$ *rejects isolated objects in* K_1 *and* K_2.

Proof. Let $S_i \in \tilde{K}_1$, S_i be an isolated object in K_1 ($1 \le i \le k$). Then, any nonempty $N_{Q(S_i, S_t)}$ ($t = 1, 2, \ldots, i-1, i+1, \ldots, k$) contains zeros of the function F_1. Therefore, any interval containing S_i and corresponding to an admissible conjunction contains no S_t from $\tilde{K}_1 \backslash S_t$. Formulas (5) and (6) show that $\Gamma_1(S_i) = 0$.

Clearly, $\Gamma_2(S_i) = 0$, since none of the elementary conjunctions in $\mathfrak{D}(F_2)$ is equal to unity on S_t. The theorem is thus proved.

Corollary. If the number of isolated in K_1 and K_2 objects from $[TO]_{n,k,l}$ equals p and d, respectively, then

$$\max_{\mathcal{A}} \varphi_{sk}(\mathcal{A}) = 1 - \frac{p+d}{k+l},$$

and this maximum is attained on any $\mathcal{A}$ with a sufficient pair of DNFs and positive parameters $W(Q)$ and $W(S_i)$.

It follows from the proof of the theorem that in order to find out the results of the sliding check by $[TO]_{n,k,l}$, it is sufficient to extract all isolated objects in K_1 and K_2. For this purpose, it is sufficient to construct not more than $k(k-1)$ nonempty $N_{Q(S_i, S_j)}$ in $\tilde{K}_1$ and not more than $l(l-1)$ nonempty $N_{Q(S_u, S_v)}$ in $\tilde{K}_2$. For every $Q(S_i, S_j)$, l equalities $Q(S_i, S_j)(S_r) = 0$ $(r = k+1, \dots, k+l)$ should be checked. Similarly, for every $Q(S_u, S_v)$, k equalities should be checked. Obviously, the total number of operations for constructing of all isolated objects does not exceed

$$\mathrm{const}\, n(lk(k-1) + kl(l-1)) = (\mathrm{const})' nkl(k+l-2).$$

It also follows from the facts proved above that arbitrary positive parameters can be used when constructing an optimal algorithm with respect to φ_{sl}. This casts doubt on the possibility of using the sliding check for estimating the quality of recognizing algorithms.

At least one sufficient pair $(\mathfrak{D}_1(F_1)\mathfrak{D}_2(F_2)$ can easily be constructed. Isolated objects are realized by arbitrary conjunctions

$$\bigvee Q_t^1 \text{ in } \mathfrak{D}_1(F_1), \quad \bigvee Q_r^2 \text{ in } \mathfrak{D}_2(F_2).$$

Then,

$$\mathfrak{D}_1(F_1) = \bigvee Q_t^1 \vee \bigvee Q(S_i, S_j) \text{ over all admissible } Q(S_i, S_j),$$
$$i = 1, 2, \dots, k, \; j = 1, 2, \dots, i-1, i+1, \dots, k,$$

$$\mathfrak{D}_2(F_2) = \bigvee Q_r^2 \vee \bigvee Q(S_u, S_v) \text{ over all admissible } Q(S_u, S_v),$$
$$u = k+1, 2, \dots, k+l, \;\; v = k+1, \dots, i-1, u+1, \dots, k+l.$$

4. OPTIMIZATION WITH RESPECT TO THE INDEPENDENT CHECK

We consider the problem of constructing an optimal algorithm with respect to the functional $\varphi_{\mathrm{in}}(\mathcal{A})$. More precisely, we seek an algorithm that yields the maximum number of correct recognition results for $m+q$ object rows $TK_{n,m,q}$. Let us write out the conditions of correct recognition for the objects $S_1', \dots, S_m'$, $S_{m+1}', \dots, S_{m+q}'$:

$$\begin{gathered} \Gamma_1(S_1') > \Gamma_2(S_1'), \dots, \Gamma_1(S_m') > \Gamma_2(S_m'), \\ \Gamma_1(S_{m+1}') > \Gamma_2(S_{m+1}'), \dots, \Gamma_1(S_{m+q}') > \Gamma_2(S_{m+q}'). \end{gathered} \tag{7}$$

According to (4), system (7), for fixed $\mathfrak{D}_1(F_1)$, $\mathfrak{D}_2(F_2)$, is generally an inconsistent system of bilinear inequalities.

The optimal (with respect to $\varphi_{\mathrm{in}}(\mathcal{A})$) algorithm is determined by (any) solution to the maximal consistent subsystem of (7). Methods for finding such subsystems are computationally very costly. Many studies are devoted to the construction of approximate methods for solving this problem (see, e.g., [7, 8]).

Assume that a consistent subsystem

$$\begin{gathered} \Gamma_1(S_{i_1}) > \Gamma_2(S_{i_1}), \dots, \Gamma_1(S_{i_p}) > \Gamma_2(S_{i_p}), \quad p \le k, \\ \Gamma_1(S_{j_1}) > \Gamma_2(S_{j_1}), \dots, \Gamma_1(S_{j_d}) > \Gamma_2(S_{j_d}), \quad d \le l \end{gathered} \tag{8}$$

has been found.

Any solution to (8) can be used for constructing the optimal algorithm. Usually, the solution is found by solving the problem

$$\left(\sum_{i=1}^{k+l} |1 - W(S_i)| + \sum_{Q \in \mathfrak{D}(F_1)} |1 - W(Q)| + \sum_{Q \in \mathfrak{D}(F_2)} |1 - W(Q')| \right) \longrightarrow \min \tag{9}$$

subject to constraints (8).

In many cases, (9) is replaced by the following functional, which can be optimized by conventional methods:

$$\left(\sum_{t=1}^{k+l}[1-W(S_t)] + \sum_{Q\in\mathfrak{D}(F_1)}[1-W(Q)] + \sum_{Q'\in\mathfrak{D}(F_2)}[1-W(Q')]\right) \longrightarrow \min, \quad (10)$$

$$0 \le W(S_t) \le 1, \quad 0 \le W(Q), \quad W(Q') \le 1.$$

This optimization procedure must be complemented with an algorithm for choosing $\mathfrak{D}_1(F_1)$ and $\mathfrak{D}_2(F_2)$, which makes the synthesis of $\mathcal{A}$ very complicated. For this reason, it is important to use methods that make it possible (if only partially) to construct DNFs for F_1 and F_2, reduce the number of inequalities in (7), and find at least some values of $W(S_t)$, $W(Q)$, and $W(Q')$ using not very costly methods.

Let $\tilde{F}_1$ and $\tilde{F}_2$ be incompletely defined Boolean functions:

$$\tilde{F}_1 = \begin{cases} 1 \text{ on } \tilde{K}_1 = \{S_1, \dots, S_k\}, \\ 0 \text{ on } \tilde{K}_2 \cup \{S'_{m+1}, \dots, S'_{m+q}\}, \text{ i.e., on the objects from } K_2 \text{ that belong} \\ \text{to } [TO]_{n,k,l} \text{ and } [TK]_{n,m,q}, \text{ undefined for the other arguments,} \end{cases}$$

$$\tilde{F}_2 = \begin{cases} 1 \text{ on } \tilde{K}_2 = \{S_{k+1}, \dots, S_{k+l}\}, \\ 0 \text{ on } \tilde{K}_1 \cup \{S'_1, \dots, S'_m + q'\}, \text{ i.e., on the objects from } K_1 \text{ that belong} \\ \text{to } [TO]_{n,k,l} \text{ and } [TK]_{n,m,q}, \text{ undefined for the other arguments.} \end{cases}$$

Let S'_j be the test object from K_1, and let there exist S_t in $\tilde{K}_1$ such that $N_{Q(S_t,S'_j)}$ does not contain zeros of the function $\tilde{F}_1$, i.e., $Q(S_t, S'_j)(S_i) = 0$ for $i = k+1, \dots, k+l$ and $Q(S'_u, S'_j = 0$ for $u = m+1, \dots, m+q$. Fix all such $S'_{j_1}, \dots, S'_{j_p}$ in K_1 belonging to $[TK]_{n,m,q}$.

Similarly, we determine and fix $S'_{u_1}, \dots, S'_{u_d}$ in K_2 belonging to $[TK]_{n,m,q}$. (Here, K_1 is replaced with K_2 and $\tilde{F}_1$ with $\tilde{F}_2$).

Set $\mathfrak{M}_1 = \{S'_{j_1}, \dots, S'_{j_p}, S'_{u_1}, \dots, S'_{u_d}\}$.

Let S'_v be a test object in $\tilde{K}_1$ and all $N_{Q(S_i,S'_v)}$ $(i = 1, 2, \dots, k)$ contain zeros of the function F_1, i.e., sets from $\tilde{K}_2$. Fix all such $S'_{v_1}, \dots, S'_{v_r}$ in K_1 belonging to $[TK]_{n,m,q}$. Similarly, we determine and fix $S'_{u_1}, \dots, S'_{u_h}$ in K_2 belonging to $[TK]_{n,m,q}$.

Set $\mathfrak{M}_2 = \{S'_{v_1}, \dots, S'_{v_r}, S'_{u_1}, \dots, S'_{u_h}\}$.

Proposition 2. *None of the elements in $\mathfrak{M}_2$ will be recognized by any algorithm with arbitrary DNFs $\mathfrak{D}_1(F_1)$ and $\mathfrak{D}_2(F_2)$ that realize incompletely defined F_1 and F_2.*

Proof. S'_{v_1} $(i = 1, 2, \dots, r)$ cannot be placed into K_1, since any interval covering S'_{v_i} and any training S_t in $\tilde{K}_1$ contain zeros of F_1. Therefore, no elementary conjunction Q such that $Q(S'_{v_i}) = 1$ can take the value 1 on any S_t from $\tilde{K}_1$. Hence, any algorithm $\mathcal{A}$ such that $\Gamma_1(S'_{v_i}) = 0$ either rejects the object S'_{v_i} or assigns it to K_2, i.e., misrecognizes it. Proposition 1 for $S'_{u_1}, \dots, S'_{u_h}$ in K_2 belonging to $[TK]_{n,m,q}$ is proved similarly.

Corollary. The inequalities corresponding to the objects $S'_{v_1}, \dots, S'_{v_r}, S'_{u_1}, \dots, S'_{u_h}$ should be removed from system (7).

Let us remove the objects belonging to the set $\mathfrak{M}_1 \cup \mathfrak{M}_2$ from the control sample. We also remove the corresponding inequalities from system (7).

We construct DNFs $\tilde{\mathfrak{D}}_1(F_1)$ and $\tilde{\mathfrak{D}}_2(F_2)$ that realize F_1 and F_2, respectively. Then, we find a consistent subsystem (possibly maximal) in the system of bilinear inequalities constructed for $\{S'_1, \ldots, S'_m, S'_{m+1}, \ldots \ldots, S'_{m+q}\} \backslash \mathfrak{M}_1 \cup \mathfrak{M}_2$ and solve it to obtain the values of the parameters $W(Q)$, $Q \in \tilde{\mathfrak{D}}_1(F_1)$, $W(Q')$, $Q' \in \tilde{\mathfrak{D}}_2(F_2)$, and $W(S_1), \ldots, W(S_{k+l})$.

Thus, an algorithm $\tilde{\mathcal{A}}$ is specified. Let $M_1(\tilde{\mathcal{A}})$ and $M_2(\tilde{\mathcal{A}})$ be the sets of objects in $[TK]_{n, m, q}$ belonging to the classes K_1 and K_2, respectively that are recognized by $\tilde{\mathcal{A}}$.

Theorem 3. *There exists an addition of conjunctions $Q(S_i, S'_j)$ $(S_i \in [TO]_{n, k, l}, S'_j \in [TK]_{n, q, m})$ to $\tilde{\mathfrak{D}}_1(F_1)$ and $\tilde{\mathfrak{D}}_2(F_2)$ such that specifying $W(Q(S_i, S'_j))$ and changing some of the parameters $W(S_i)$ $(i = 1, 2, \ldots, k + l)$, one can design an algorithm $\mathcal{A}$ that recognizes all objects in $\mathfrak{M}_1$ and such that*

$$M_1(\tilde{\mathcal{A}}) \subseteq M_1(\mathcal{A}), \quad M_2(\tilde{\mathcal{A}}) \subseteq M_1(\mathcal{A}).$$

Proof. Let $S'_j \in K_1$ $(1 \le j \le m)$. By the definition of M_1, $[TO]_{n, k, l}$ contains an $S_i \in \tilde{K}_1$ $(1 \le i \le k)$ such that $N_{Q(S_i, \tilde{S}_j)}$ does not contain zeros of the function $\tilde{F}_1$. Therefore, $Q(S_i, \tilde{S}_j)$ is an admissible elementary conjunction for F_1, and $N_{Q(S_i, \tilde{S}_j)}$ does not contain test objects from K_2. We also assume that $\Gamma_1(S'_j) \le \tilde{\Gamma}_2(S'_j)$ in $\tilde{\mathcal{A}}$, i.e., the object S'_j is not assigned to K_1 by the algorithm $\mathcal{A}$. Otherwise, we pass to the next object in $\mathfrak{M}_1$ without changing $\mathcal{A}$. Let us select all $S_{u_1}, \ldots, S_{u_p}$ in $\tilde{K}_1$ such that $Q(S_i, \tilde{S}_j)(S_{u_r}) \neq 0$ for $r = 1, 2, \ldots, p$. Consider two cases.

Case 1. Among $W(S_{u_r})$ $(r = 1, 2, \ldots, p)$, there exists at least one $W(S_u) \neq 0$ $(1 \le h \le p)$ in $\tilde{\mathcal{A}}$. Set

$$W(Q(S_i, S'_j)) = \frac{\tilde{\Gamma}_2(S'_j) - \tilde{\Gamma}_1(S'_j) + c}{W(S_{u_h})}, \quad c > 0. \tag{11}$$

Let us add the elementary conjunction $Q(S_i, S'_j)$ with the weight $W(Q(S_i, S'_j))$ determined by (11) to $\tilde{\mathfrak{D}}_1(F_1)$; the other parameters remain unchanged. The new algorithm is denoted by $\tilde{\mathcal{A}}(+S'_j)$. In this algorithm, a quantity not less than $W(S_{u_h})$ is added to the estimate $\tilde{\Gamma}_1(S'_j)$: $W(Q(S_i, S'_j)) = \tilde{\Gamma}_2(S'_j) - \tilde{\Gamma}_1(S'_j) + c$. The estimate $\tilde{\Gamma}_2(S'_j)$ does not change in $\tilde{\mathcal{A}}(+S'_j)$, since $Q(S_i, S'_j)$ is not involved in the construction of estimates for K_2. Therefore, the object S'_j will be correctly assigned by the algorithm $\tilde{\mathcal{A}}(+S'_j)$ to the class K_1. Objects from M_1 can only increase their estimates voting for K_1, and the estimates voting for K_2 will remain unchanged. Therefore, $M_1(\tilde{\mathcal{A}}) \subseteq M_1(\tilde{\mathcal{A}}(+S'_j))$. The interval $N_{Q(S_i, S_j)}$ does not overlap the test objects from K_2. Therefore, both estimates remain unchanged for these objects; i.e., $M_2(\tilde{\mathcal{A}}) \subseteq M_2(\tilde{\mathcal{A}}(+S'_j))$.

Case 2. $W(S_{u_r}) = 0$ in $\tilde{\mathcal{A}}$ $(r = 1, 2, \ldots, p)$. Let

$$d = \min_{S_v \in M_2(\tilde{A})} [\tilde{\Gamma}_2(S'_v) - \tilde{\Gamma}_1(S'_v)]. \tag{12}$$

Set $W(S_{u_1}) = \varepsilon$ and $W(S_{u_i}) = 0$ for $i = 2, 3, \ldots, r$. Let us select in $\tilde{\mathfrak{D}}_1(F_1)$ all elementary conjunctions $Q_{t_1}, \ldots, Q_{t_p}$ such that $Q_{t_j}(S_{u_1}) = 1$. Let $W(Q_{t_1}) + \ldots + W(Q_{t_p}) = N$. Here, $W(Q_{t_i})$ is a parameter of the algorithm $\tilde{\mathcal{A}}$. We choose ε so as to ensure the inequality

$$\varepsilon(W(Q_{t_1}) + \ldots + W(Q_{t_p})) < d.$$

Then, the transition from $\tilde{\mathcal{A}}$ to $\tilde{\mathcal{A}}(+S'_j)$ is made as in case 1. It is clear that the new algorithm will assign S'_j to the class K_1 (i.e., it will be recognized), and $M_1(\tilde{\mathcal{A}}) \subseteq M_1(\tilde{\mathcal{A}}(+S'_j)) < d$. However, if we take into account (12), then the estimate of voting for K_2 will remain greater than that for K_1; i.e., $M_2(\tilde{\mathcal{A}}) \subseteq M_2(\tilde{\mathcal{A}}(+S'_j))$.

The completion of the algorithm is carried out sequentially for the objects from $K_1 \cap \mathfrak{M}_1$. Then (quite similarly), it is carried out for the objects from $\mathfrak{M}_1 \cap K_2$. Obviously, at this stage, the conjunctions $Q(S_u, S'_v)$, $S'_v \in \mathfrak{M}_1 \cap K_2$ are added to $\tilde{\mathfrak{D}}_1(F_2)$. The theorem is proved.

Note that, instead of $Q(S_t, S'_j)$, one could add the conjunctions Q_t such that $N_{Q(S_t, S'_j)} \subset N_{Q_t}$ and the intervals N_{Q_t} do not contain zeros of $\tilde{F}_1$ (respectively, $\tilde{F}_2$) if $S'_j \in K_1$ (respectively, $S'_j \in K_2$). The computation of the weights of the new conjunctions and the replacement of certain zero $W(S_t)$ with positive values is made in the same way as in the proof of Theorem 3.

It is easily seen that the selections of objects in $\mathfrak{M}_2$ and the completion of the algorithm $\tilde{\mathcal{A}}$ so as to make it recognize all objects in $\mathfrak{M}_1$ is not computationally costly. If $\chi(\mathcal{A}, S)$ is the number of operations used to recognize S by $\mathcal{A}$, then the total cost of constructing $\mathcal{A}$ that does not make errors on $\mathfrak{M}_1$ and is no less accurate than $\tilde{\mathcal{A}}$ on $[TK]_{n,m,q} \backslash (\mathfrak{M}_1 \cup \mathfrak{M}_2)$ does not exceed a quantity proportional to $\chi(\mathcal{A})|\mathfrak{M}_1|$. At the same time, the elimination of $|(\mathfrak{M}_1 \cup \mathfrak{M}_2)|$ inequalities from the system simplifies the construction of an optimal or approximately optimal algorithm with respect to φ_{in}.

5. A TECHNIQUE FOR INCREASING THE ACCURACY OF RECOGNITION ON $[TK]_{n,m,q}$

Assume that all operations described in Section 4 have been performed in the problem with $[TO]_{n,k,l}$ and $[TK]_{n,m,q}$. Assume that an algorithm $\mathcal{A}$ has been constructed that have not correctly recognized the objects $S'_{i_1}, \ldots, S'_{i_d}$ $(d \le m)$ in K_1 and $S'_{j_1}, \ldots, S'_{j_h}$ $(d \le m)$ in K_2.

Let us introduce incompletely defined functions $\hat{F}_1$ and $\hat{F}_2$: $\hat{F}_1(S_i) = 1$ $(i = 1, 2, \ldots, k)$, $\hat{F}_1(S_j) = 0$ $(j = k+1, \ldots, k+l)$, and $\hat{F}_1(S'_t) = 0$ for all S'_t in $\{S'_{m+1}, \ldots, S'_{m+q}\} \backslash \{S'_{j_1}, \ldots, S'_{j_h}\}$. For all other arguments in E^n, the function $\hat{F}_1$ is undefined. In contrast to $\hat{F}_1$, the function $\hat{F}_2$ is equal to zero not for all objects in K_2, but only for those of them that are recognized by the algorithm $\mathcal{A}$. More precisely, $\hat{F}_2(S_j) = 1$ for $j = k+1, \ldots, k+l$, $\hat{F}_2(S_i) = 0$ for $i = 1, 2, \ldots, k$, and $\hat{F}_2(S'_v) = 0$ for all S'_v in $\{S'_1, \ldots, S'_m\} \backslash \{S'_{i_1}, \ldots, S'_{i_d}\}$, i.e., for all correctly recognized test objects in K_1. Finally, $\hat{F}_2$ is undefined for all other arguments.

Consider any S'_{i_j} $(1 \le j \le d)$ in the set $\{S'_{i_1}, \ldots, S'_{i_d}\} \subseteq K_1$ for which there exists an $S_t \in K_1$ such that

$$N_{Q(S_t, S'_{i_j})} \cap N_0(\hat{F}_1) = \varnothing. \tag{13}$$

In contrast to the constructs in Section 4, $Q(S_t, S'_{i_j})$ can be equal to 1 on test objects from K_2, but only on those of them that were not assigned to K_2 by the algorithm described in Section 4. Again, assume that the algorithm $\mathcal{A}$ has correctly recognized the test objects in the sets $M_1(\mathcal{A})$ and $M_2(\mathcal{A})$ belonging to the classes K_1 and K_2 in the test table, respectively. Let the function F_1 in $\mathcal{A}$ be implemented by the DNF $\mathfrak{D}_{\mathcal{A}}(F_1)$.

We add to $\mathfrak{D}_{\mathcal{A}}(F_1)$ the elementary conjunction $Q(S_t, S'_{i_j})$ (or any conjunction that absorbs the former conjunction, whose interval is admissible for F_1 and does not contain test objects from K_2 that were recognized by $\mathcal{A}$). Then, we carry out the constructions similar to those used in the proof of Theorem 3. If $W(S_t) \neq 0$ in $\mathcal{A}$, then we define $W(Q(S_t, S'_{i_j}))$ so as to ensure the inequality $\Gamma_1(S'_{i_j}) > \Gamma_1(S'_{i_j})$ in the new algorithm $\tilde{\mathcal{A}}$.

In $\mathcal{A}$, we have $\Gamma_2 > \Gamma_1$ for S'_{i_j}. It is sufficient to set $W(Q(S_t, S'_{i_j}))\ W(S'_{i_j}) > \Gamma_2 - \Gamma_1$. If the elementary conjunction $W(Q(S_t, S'_{i_j}))$ is already included in $\mathfrak{D}_{\mathcal{A}}(F_1)$, then the weight W should be increased by a quantity c such that the following inequality be satisfied:

$$cW(S_t) > \Gamma_2 - \Gamma_1.$$

It is easy to see that the estimate Γ_1 for the objects in $\tilde{K}_1$ will not be less. The estimates of the recognized objects in K_2 will remain unchanged, since $Q(S_t, S'_{i_j})$ is zero for them.

If $W(S_t) = 0$ and $Q(S_t, S_{i_j})$ is not included in $\mathfrak{D}_{\mathcal{A}}(F_1)$, we set $W(S_t) = \varepsilon$ as in the proof of Theorem 3. ε is chosen so small that the increase of the estimate Γ_1 for recognized test objects in K_2 (the set $M_2(\mathcal{A})$) would retain the inequality $\Gamma_2 > \Gamma_1$ for these objects. Then, as in the previous case, we set the value for $W(S_t)$.

Finally, if $W(S_t) = 0$ in $\mathcal{A}$ and $Q(S_t, S'_{i_j}) \in \mathfrak{D}_{\mathcal{A}}(F_1)$, we set, as before, $W(S_t) = \varepsilon$ so as to retain the correct recognition of objects in K_2, and add the quantity c to $W(Q(S_t, S'_{i_j}))$ so as to ensure the inequality $\Gamma_1(S'_{i_j}) > \Gamma_2(S'_{i_j})$.

Thus, we construct the algorithm $\tilde{\mathcal{A}}$ for which $M_2(\tilde{\mathcal{A}}) = M_2(\mathcal{A})$ and $M_1(\mathcal{A} \cup S'_{i_j} \subseteq M_1(\tilde{\mathcal{A}}))$.

Obviously, we could take for S'_{i_j} any misrecognized object in $\tilde{K}_1$ satisfying (13).

The algorithm $\mathcal{A}$ is completed quite similarly for the misrecognized object S'_{u_v} in $\tilde{K}_2$ if there exists an $S_i \in \tilde{K}_2$ such that

$$N_{Q(S_i, S'_{u_v})} \cap N_0(\hat{F}_2) = \varnothing.$$

In this case, the elementary conjunction $Q(S_i, S'_{u_v})$ is added to $\mathfrak{D}_{\mathcal{A}}(F_2)$ (if it is not already included in $\mathfrak{D}_{\mathcal{A}}(F_2)$). All assignments and changes of the parameters $W(Q(S_i, S'_{u_v}))$ and $W(S_t)$ are carried out according to the same rules, and the sets $M_1(\mathcal{A})$ and $M_2(\mathcal{A})$ interchange their roles in the description of the construction.

It is clear that the correction of $\mathcal{A}$ can be carried out several times. Generally, the result depends on the order of including objects from $\tilde{K}_1$ and $\tilde{K}_2$ satisfying condition (13) into $M_1(\mathcal{A})$ and $M_2(\mathcal{A})$, respectively. Upon every inclusion, the domain of definition of one of the functions $\hat{F}_1$ or $\hat{F}_2$ is extended.

6. ON CONJUNCTIONS THAT ABSORB THE CONJUNCTIONS $Q(S_t, S'_j)$

When the DNFs $\mathfrak{D}(F_1)$ and $\mathfrak{D}(F_2)$ are constructed, it is often useful to include into them not only maximum-rank conjunctions, which are equal to 1 at the points S_t and S'_j, but also absorbing conjunctions.

In the process, the extension of intervals should not result in a nonempty intersection not only with zeros of the functions F_1 and F_2, respectively, but also with the selected subsets of test objects in the classes K_2 and K_1, respectively. Earlier, the selected subsets consisted either of all objects in K_2 and K_1 or the objects in these classes that were recognized by the algorithm $\mathcal{A}$.

Let us formulate a more general problem. The following data are given: an object S_i in $\tilde{K}_1$ (in $\tilde{K}_2$, respectively) such that $S_t \in [TO]_{n,k,l\times 2}$; a subset K'_1 (respectively, K'_2) in $[TK]_{n,m,q}$, $|K'_1| \le m$, $|K'_2| \le q$; a subset K''_2 (respectively, K''_1) of test objects in K_2 (in K_1, respectively).

It is required to construct all maximal intervals N_Q containing all points of the set $S_t \cup K'_1$ (respectively, $S_i \cup K'_2$) and having no common points with the set K''_2 (K''_1, respectively).

We solve this problem for $S_t \in \tilde{K}_1$, K'_1, and K''_2. The solution for $S_i \in \tilde{K}_2$, K'_2, and K''_1 is completely similar.

Let $\tilde{\alpha}^t = S_t = \{\alpha_{t1}, \ldots, \alpha_{tn}\}$, $K_1' = \{\tilde{\beta}^{u_1}, \ldots, \tilde{\beta}^{u_d}\}$ ($\tilde{\beta}^{u_i} = S'_{u_i}$, $i = 1, 2, \ldots, d$, $1 \le u_1, \ldots, u_d \le m$), and $K_2' = \{\tilde{\beta}^{v_1}, \ldots, \tilde{\beta}^{v_h}\}$, $\tilde{\beta}^{v_j} = S'_{u_j}$, $j = 1, 2, \ldots, h$, $m + 1 \le v_1, \ldots, v_h \le m + q$.

Let us find all identical columns in the matrix consisting of the rows $\tilde{\alpha}^t$, $\tilde{\beta}^{u_1}, \ldots, \tilde{\beta}^{u_d}$. Let their indexes be $t_1, \ldots, t_l$. Then (see Proposition 1), the minimal interval containing $S_t \cup K_1'$ is

$$N_{x_{t_1}^{\alpha_{tt_1}} \ldots x_{t_l}^{\alpha_{tt_l}}} = N_{Q(S_t, K_1)}.$$

Evidently, all required $\tilde{Q}$ have intervals that contain Q. Note that if the matrix consisting of the rows $\tilde{\alpha}^t$, $\tilde{\beta}^{u_1}, \ldots, \tilde{\beta}^{u_d}$ contains no identical columns or $N_{Q(S_t, K_1')} \cap K_2'' \ne \varnothing$, then this problem has no solutions. Otherwise, the set of required conjunctions is nonempty.

For every $\tilde{\beta}^{v_j}$ ($j = 1, 2, \ldots, h$), we find in the set $\{t_1, \ldots, t_l\}$ all indices $t_{r_1}, \ldots, t_{r_p}$ ($p \le l$) such that $\alpha_{tt_{r_1}} \ne \beta_{v_j t_{r_1}}, \ldots, \alpha_{tt_{r_p}} \ne \beta_{v_j t_{r_p}}$.

Let us associate with the factors $Q(S_t < K_1')$ the variables $y_{t_1}, \ldots, y_{t_l}$. In order that $\tilde{\beta}^{v_j} \,\overline{\in}\, N_{\tilde{Q}}$, it is necessary and sufficient that $\tilde{Q}$ include at least one factor with the associated $y_{t_{r_1}}, \ldots, y_{t_{r_p}}$; i.e.,

$$\mathfrak{D}(S_t, K_1', \tilde{\beta}^{v_j}) = y_{t_{r_1}} \vee \ldots \vee y_{t_{r_p}} = 1, \quad j = 1, 2, \ldots, h.$$

Reducing

$$\prod_{j=1}^{p} \mathfrak{D}(S_t, K_1', \tilde{\beta}^{v_j})$$

to the DNF and simplifying y_t ($y_t = y_t$, $Q \vee QQ' = Q$), we obtain the DNF $\tilde{\mathfrak{D}} = \bigvee y_{t_{r_1}} \ldots y_{t_{r_g}}$. Every conjunction in this DNF has a corresponding required conjunction

$$x_{t_{r_1}}^{\alpha_{tt_{r_1}}} \ldots x_{t_{r_g}}^{\alpha_{tt_{r_g}}}$$

and every required conjunction has a corresponding conjunction in $\tilde{\mathfrak{D}}$. The simple proof is conducted by conventional techniques. In total, the multiplication of $|K_2''| \le q$ parentheses is required. In the corresponding problem for $S_t \in \tilde{K}_2$, the number of parentheses to be multiplied is $|K_1''| \le m$.

For large q and (or) m, such a reduction to the DNF is very costly. Indeed, the problem is reduced to finding all upper zeros of a monotone Boolean function with the number of variables equal to the number of factors in $Q(S_t, K_1')$.

With the factors $x_{t_1}^{\alpha_{tt_1}}, \ldots, x_{t_l}^{\alpha_{tt_l}}$, we associate the Boolean variables $y_1, \ldots, y_l$. Define the function $f(x_1, \ldots, x_l)$ on E^l in the following way. Let $\tilde{\gamma} = (\gamma_1, \ldots, \gamma_l)$, $\gamma_{i_1}, \ldots, \gamma_{i_d} = 1$, and all other components be zero. We associate with $\tilde{\gamma}$ the conjunction $Q(i_1, \ldots, i_d)$ consisting of the factors $Q(S_t, K_1')$ with the indexes $i_1, \ldots, k_d$:

$$f(\tilde{\gamma}) = \begin{cases} 1, & \text{if } N_{Q(i_1, \ldots, i_d)} \cap K_2'' \ne \varnothing, \\ 0 & \text{otherwise.} \end{cases}$$

Proposition 3. *The function $f(x_1, \ldots, x_l)$ is monotone. There exists a one-to-one correspondence described in the definition of f between the lower zeros of f and the conjunctions that are simple implicants for $\hat{F}_1$ and absorb $Q(S_t, K'_1)$.*

The proof is simple and is conducted by conventional methods. In order to construct maximal extensions of the interval $N_{Q(S_t, K'_1)}$, any of the numerous methods for deciphering monotone Boolean functions can be used.

To construct the parts of $\mathfrak{D}(F_1)$ and $\mathfrak{D}(F_2)$ consisting of conjunctions whose intervals contain simultaneously training and test objects from the corresponding class and do not contain the selected subset of objects of another class, one should perform the following operations (for definiteness, we consider the class K_1):

(1) For every S_t ($S_t \in K_1$, $S_t \in [TO]_{n,k,l}$), subsets of test objects $M(S_t, K_1)$ in K_1 are selected such that there exists an interval N_Q satisfying the condition

$$S_t \cup M(S_t, K_1) \subseteq N_Q;$$

(2) Among all intervals N_Q, those having an empty intersection with the training objects in K_2 are selected.

If these operations are cannot be executed because of their complexity, one can restrict oneself to a part of them (generally, not necessarily maximal).

For example, all nonempty $N_{Q(S_i, S'_j)}$ ($i = 1, 2, \ldots, k, j = 1, 2, \ldots, m$) are constructed that have a nonempty intersection with the training objects in K_2 and the selected subset of control objects in K_2 (maybe empty). The subsets containing S_j for which $N_{Q(S_i, S'_j)}$ does not satisfy these conditions are not considered further. Then, a similar selection of the triples (S_i, S'_u, S'_v) is performed, the number of which is evidently not greater than $km(m-1)$, and so on.

The problem can be easily reduced to deciphering the corresponding monotone Boolean function.

ACKNOWLEDGMENTS

This work was supported by the Russian Foundation for Basic Research (project nos. 99-01-00433 and 00-15-96064), "Intelligent Computer Technologies Program" of the Russian Academy of Sciences, and INTAS program, project no. 00-397.

REFERENCES

1. Zhuravlev, Yu.I., Set-Theoretic Methods in Boolean Algebra, *Probl. Kibern.*, issue 8, Moscow: Nauka, 1962.
2. Zhuravlev, Yu.I., Algorithms for Constructing Minimal Disjunctive Forms for Functions in Boolean Algebra, *Diskretnaya matematika i matematicheskie voprosy kibernetiki* (Discrete Mathematics and Mathematical Problems of Cybernetics), Moscow: Nauka, 1974., pp. 67–98.
3. Zhuravlev, Yu.I., *Izbrannye nauchnye trudy* (Selected Research Works), Moscow: Magistr, 1998.
4. Zhuravlev, Yu.I., An Algebraic Approach to Recognition and Classification Problems, *Probl. Kibern.*, issue 33, Moscow: Nauka, 1978.
5. Zhuravlev, Yu.I., Correct Algebras over Sets of Incorrect (Heuristic) Algorithms: Part I, *Kibern.*, 1977, no. 4, pp. 5–17.
6. Baskakova, L.V. and Zhuravlev, Yu.I., A Model of Recognizing Algorithms with Representative Sets and Systems of Support Sets, *Zh. Vychisl. Mat. Mat. Fiz.*, 1981, vol. 21, no. 5, pp. 1264–1275.
7. Ryazanov, V.V., On the Construction of Optimal Recognition and Taxonomy Algorithms in Application Problems, *Raspoznavanie, klassifikatsiya i prognoz. Matematicheskie metody i ikh primenenie* (Recognition, Classification, and Forecast: Mathematical Methods and Their Application), Moscow: Nauka, 1988, issue 1, pp. 229–279.
7. Ryazanov, V.V., On the Optimization of a Class of Recognition Models, *Pattern Recognit. Image Anal.*, 1991, vol. 1, no. 1, pp. 108–118.